Model-Based Tracking at 300Hz using Raw Time-of-Flight Observations

Jan Stühmer [§†] Sebastian Nowozin [†] Andrew Fitzgibbon [†] Richard Szeliski [†]
Travis Perry [‡] Sunil Acharya [‡] Daniel Cremers [§] Jamie Shotton [†]

[§]Technische Universität München [†]Microsoft Research [‡]Microsoft

Abstract

Consumer depth cameras have dramatically improved our ability to track rigid, articulated, and deformable 3D objects in real-time. However, depth cameras have a limited temporal resolution (frame-rate) that restricts the accuracy and robustness of tracking, especially for fast or unpredictable motion. In this paper, we show how to perform model-based object tracking which allows us to reconstruct the object's depth at an order of magnitude higher frame-rate through simple modifications to an off-the-shelf depth camera. We focus on phase-based time-of-flight (ToF) sensing, which reconstructs each low frame-rate depth image from a set of short exposure 'raw' infrared captures. These raw captures are taken in quick succession near the beginning of each depth frame, and differ in the modulation of their active illumination. We make two contributions. First, we detail how to perform model-based tracking against these raw captures. Second, we show that by reprogramming the camera to space the raw captures uniformly in time, we obtain a 10x higher frame-rate, and thereby improve the ability to track fast-moving objects.

1. Introduction

Tracking objects that move is a fundamental computer vision task that enables higher-level reasoning about the world. There are several key challenges for visual object tracking that limit the accuracy of current systems: (i) changing object appearance due to object translation, rotation, deformation, and lighting variation; (ii) occlusion, and objects leaving the viewing volume; (iii) simultaneous tracking of multiple objects whose number may vary over time; and (iv) tracking under fast object or camera motion.

General purpose tracking approaches address these challenges through adaptive non-parametric methods, as in mean-shift tracking [4], and by online learning of a flexible object representation [22]. For multiple objects longer temporal reasoning is required to disambiguate different objects with similar appearance [21, 1]. Despite significant progress in the last decade, general purpose tracking remains challenging, as illustrated by the recent VOT 2014 challenge [16]. For other general surveys on object tracking, see [27, 26].

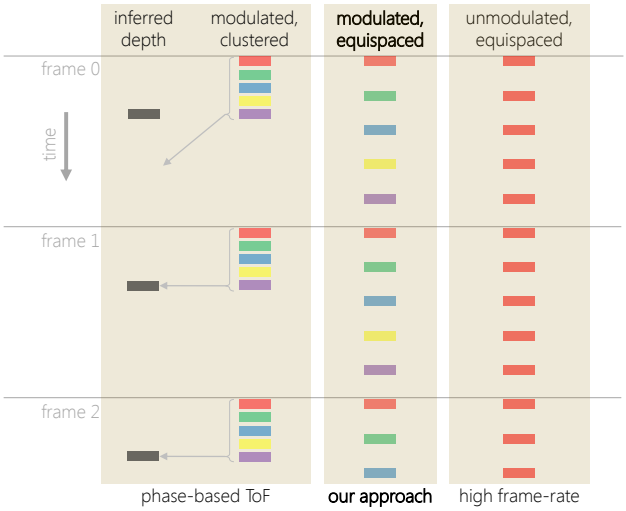

Figure 1. **Overview.** Phase-based time-of-flight (ToF) sensors infer a low frame-rate stream of depth images from a set of short-exposure 'raw' captures that are clustered closely in time to reduce motion artifacts. For illustration purposes, we use five colors here to indicate different frequencies and phase modulations of the illuminant and sensor; see text. For the application of model-based tracking, we propose to forego the depth reconstruction step, and instead track directly from *equispaced* raw captures, giving us signal at much higher frame-rate.

In this paper we do not address the general purpose tracking problem but instead focus on accurately tracking *fast-moving* rigid objects. One strategy to perform such tracking is to use custom high speed sensors and processing units [20]. Alternatively one could increase the resolution of the captured frames in order to improve angular accuracy, a point made in an extensive synthetic SLAM study [10]. A third option may be to use multiple camera systems [17].

However, increasing the frame rate via a high frame rate camera or even a custom built imaging sensor is expensive and requires control over the imaging setup. In this paper we instead show how the Kinect V2 time-of-flight (ToF) sensor [2] can be repurposed for high speed object tracking. Our approach should readily extend to other commercially available ToF sensors.

Our approach is based on the internal workings of phase-based ToF sensors, as illustrated in Fig. 1. More detail is given below, but briefly, the Kinect camera captures a set of actively illuminated infrared frames (the colored bars in the left column) and infers from these a depth image frame (each grey bar). In order to provide a 30Hz depth signal, the Kinect sensor internally captures infrared frames at an average of 300Hz. The frames are captured in a short burst at the beginning of the frame, to minimize movement in the scene during the capture period required for depth reconstruction. Each frame is captured under one of three frequencies of laser illumination, modulated by one of three phases, resulting in nine infrared frames per depth frame. A tenth infrared frame without active illumination is recorded to adjust for ambient brightness independent of the active illumination.

Our main contribution is to show how to repurpose this phase-based ToF sensor in two ways. First, we show that model-based tracking can be performed against the raw ToF captures. This has the nice property that we do not need to run through the noisy and potentially computational expensive depth reconstruction process in order to track. Second, instead of capturing the raw frames in bursts, we space the frames out equally in time at 300Hz. This allows for tracking much faster-moving objects than would have been possible using the original 30Hz depth reconstruction (see Fig. 2).

We report an initial study of the above ideas based on a model-based tracker. This employs a probabilistic state space model with a standard temporal prior. As the observation likelihood, we describe a generative model that allows us to compare the observation to a rendered simulation of the raw ToF captures, given a 3D model. Further, we show that we can accurately track a fast moving rigid object (> 6 m/s) in the regime where the depth reconstruction fails.

2. Background

In this section we present some background material on phase-based ToF and model-based tracking that will be necessary to explain our main contributions in the subsequent sections.

2.1. Phase-modulation time-of-flight

Modern ToF cameras operate based on the principle of phase modulation: a modulated light source emits a sinusoidal light signal at a specific frequency, and a special sensor images the light's reflection, gain-modulated at the same frequency [23, 18]. By recording a large number of periods during the frame exposure time the recorded image intensities contain information about the phase shift between emitted light and incoming light. This phase shift is dependent on depth but will wrap around several times with the depth ranges present in typical scenes. Instead of recording only a single frame at a single frequency, modern cameras therefore record a sequence of frames at multiple modulation frequencies and phase shifts. The set of recorded frames then allows unique disambiguation of surface distances based on phase unwrapping algorithms [11, 19, 9]. This standard operation model is illustrated in the leftmost column in Fig. 1.

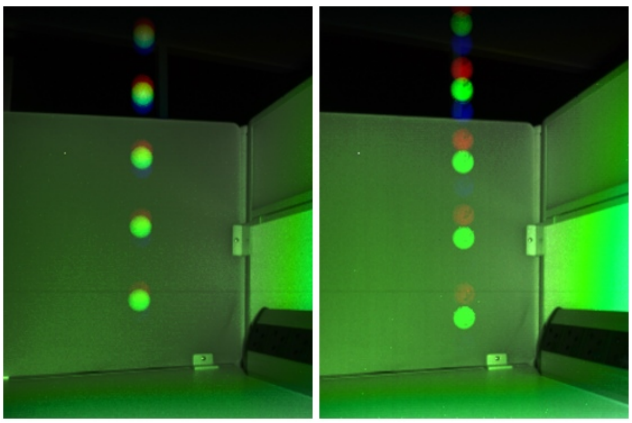

Figure 2. **Time-of-flight captures and fast motion.** A table tennis ball is dropped, and three raw captures are superimposed for visualization using the red, green, and blue color channels. **Left:** phase-based ToF clusters its captures temporally to minimize motion artifacts during depth reconstruction. **Right:** we propose reprogramming the ToF capture profiles to more equally space the captures. This paper demonstrates how to exploit this extra temporal information by tracking objects without a depth reconstruction.

Formally, for each pixel we obtain a sequence of nine measurements $R_1, \ldots, R_9$ (3 frequencies $\times$ 3 phases) via

$$R_i = \frac{\rho}{d^2} S_i(d) + \epsilon_i, \tag{1}$$

where $d > 0$ is the depth of the imaged surface at that pixel and $\rho > 0$ is the surface *albedo*. The ideal responses are dependent on modulation frequency and phase delay and are given by an idealized calibrated response curve [2],

$$S_i : [d_{\text{min}}, d_{\text{max}}] \to \{-I_{\text{max}}, \ldots, -1, 0, 1, \ldots, I_{\text{max}}\},$$

where d_{min} and d_{max} is the range of valid depths and the range of S_i are signed image intensities. For the noise model ϵ_i we simply assume zero mean Gaussian noise of a fixed standard deviation.[1]

[1] Due to the way the Kinect sensor operates [2] the right noise model would be an intensity-dependent Skellam noise, but for simplicity we adopt the Gaussian approach.

In the regular Kinect ToF sensor, a depth reconstruction engine is used to infer the depth from the nine measurements as

$$\hat{d} = f(R_1, \ldots, R_9). \tag{2}$$

Our system instead uses the raw measurements as detailed below, without first needing to infer depth.

2.2. Model-based tracking

We focus on the task of model-based object tracking [25, 15], using a generative observation model to relate the tracked position to the observations over time. To provide stable tracking we use a temporal model and follow the influential work by Isard and Blake [12] based on *particle filtering* [8] in state space models. In a state space model we need to specify a state space and both a probabilistic transition model and a probabilistic observation model [6]. We use a state vector

$$X_t = (x_t, v_t), \tag{3}$$

encoding a 3D world location $x_t \in \mathbb{R}^3$ and a 3D velocity vector $v_t \in \mathbb{R}^3$. For general rigid objects we could include rotation parameters, *i.e.* $X_t = (x_t, r_t, v_t)$, but, to demonstrate our key contributions in as simple a setup as possible, we will only use a spherical object (a table tennis ball) in the experiments, which does not require rotational parameters. While not currently demonstrated, our approach is general and our results should straightforwardly extend to more complex rigid and non-rigid objects that have higher-dimensional state spaces.

The stochastic transition model is specified via a distribution $P(X_{t+1}|X_t)$ that encodes the assumed laws of motion. The observation model is specified via an analysis-by-synthesis approach: observation Y_t corresponds to an entire raw ToF frame, and therefore we compute an observation likelihood by comparing the observed image to a synthetic rendering of the scene; we provide further details below.

Together, the transition and observation model give a joint distribution over the entire sequence of states $X_{1:T}$ and observations $Y_{1:T}$ as

$$P(X_{1:T}, Y_{1:T}) = \prod_{t=1}^{T} P(X_t|X_{t-1})\, P(Y_t|X_t), \tag{4}$$

where $P(X_1|X_0) = P(X_1)$ is assumed given.

Once the model is in place, inference given observations can be done either by *filtering* or by *smoothing* [6]. In filtering the past observations are used to infer the current believed distribution over positions and velocities. As such filtering is *causal* and suitable for interactive tracking. Filtering provides as output at each time step t the marginal distribution $P(X_t|Y_{1:t})$ over the current state X_t. In smoothing, we instead use observations both from the past and the future, *i.e.* we perform inference offline after the entire sequence $Y_{1:T}$ of T frames has been observed. This is known to significantly improve tracking accuracy [13] as the inference result $P(X_{1:T}|Y_{1:T})$ now integrates all observations coherently. A middle ground between filtering and smoothing is to delay inference by a small number of K frames and perform partial smoothing using a truncated sequence, *i.e.* to infer $P(X_{(t-K+1):t}|Y_{1:t})$. This is known as *fixed-lag smoothing* and offers an adjustable tradeoff between the two extremes [5]: for $K = 1$ we recover filtering, and for $K = T$ we recover smoothing. This can allow for improved accuracy of interactive tracking at the expense of introducing a fixed latency.

In this work we decided to use forward filtering and leave a comparison to smoothing methods for future work. We use a standard inference method: the bootstrap particle filter [8]

3. Method

We now describe our model for tracking fast moving objects. While the motion model is standard, the observation model for raw ToF captures is a novel contribution.

3.1. Motion model $P(X_{t+1}|X_t)$

Using the state representation (3) we model the motion linearly via a multivariate Gaussian distribution,

$$P(X_{t+1}|X_t) \sim \mathcal{N}\left(\begin{bmatrix} x_t + \Delta v_t \\ v_t \end{bmatrix}, \begin{bmatrix} \sigma_x^2 I_3 & 0 \\ 0 & \sigma_v^2 I_3 \end{bmatrix}\right), \tag{5}$$

where Δ is the difference in time stamps between the frame captured at step $t+1$ and step t, and $\sigma_x > 0$ and $\sigma_v > 0$ are the noise terms for the position and velocity vectors. In our experiments we set $\sigma_x = 10$ mm and $\sigma_v = 1$ mm per 300 Hz. Intuitively we can understand the model (5) as simply predicting the position x_{t+1} to be the linear extrapolation of the current position x_t using the current estimate of the velocity v_t. The velocity is assumed to remain constant, *i.e.* $v_{t+1} = v_t$, which is a common simplifying assumption. Other motion models are of course possible; we chose (5) as probably the simplest possible model that could help demonstrate our main contributions in raw ToF-based tracking.

3.2. Observation model $P(Y_t|X_t)$ for raw ToF

This section describes one of our contributions: how to create an observation model which removes the dependence on a ToF depth reconstruction and instead compute observation likelihoods directly against the raw ToF captures.

The observation model is specified as $P(Y_t|X_t)$, where Y_t is an observed raw ToF frame of size 512-by-424 (plus some meta information from the Kinect sensor), and X_t is an object hypothesis. The raw ToF frame takes the form of raw responses (1), one for each sensor element (sensel). Let

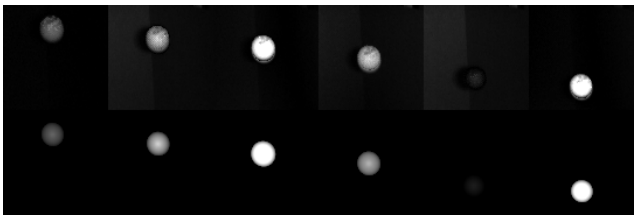

Figure 3. **Model Based Tracking.** Depending on the frequency and phase configuration of the individual exposures, the object appears with different illuminations in the raw ToF captures. The generative forward model allows to synthesize the appearance of the object for these different illuminations. First row: Observed raw ToF image. Second row: Rendered image of the best hypothesis. Columns correspond to individual exposures.

use denote by $\bar{R}_i(u)$ the observed response at sensel location u and shutter type i, where for a single frame only one shutter type i is possible. The observed information is then $Y_t = (i, \bar{\mathbf{R}}_i)$, where i is the shutter type, and $\bar{\mathbf{R}}_i$ is the vector of all response at all sensels. The shutter type i changes in a fixed cyclic order on the Kinect device, and thus does not need to be modeled probabilistically. Therefore we only need to specify a model for the frame $\bar{\mathbf{R}}_i$.

Our probabilistic model for $\bar{\mathbf{R}}_i$ is based on a 3D rendering approach: given the object hypothesis X_t we first render the distance $d(u)$ and reflectivity $\rho(u)$ for every sensel ray at location u. The reflectivity is computed via a Blinn-Phong model [3] whose coefficients we fit empirically to the object appearance prior to tracking in an offline setup step. From $d(u)$ and $\rho(u)$ and from the known shutter type i we use equation (1) to compute the expected ideal object response $R_i^{\text{obj}}(u)$ for each location u. Fig. 3 shows pairs of observed and rendered responses side by side.

The ideal response $R_i(u)$ is compared to the observed response $\bar{R}_i(u)$ to compute a likelihood. Here, a complication arises: the rendering model expects a non-zero response only at object locations, hence the background is not modeled. One possibility is to compare only sensels at the assumed object location provided by X_t, however, this does not provide a valid distribution $P(Y_t|X_t)$ for the entire observed frame.

To overcome this difficulty, we explicitly model the background. This is commonly done for RGB images via mixture models, as in the seminal work [7, 24, 28]. Here, for simplicity and because we will assume a static camera, we will use a simpler Gaussian model as follows. For every shutter type i and every location u we capture a few seconds of static background video and compute the empirical mean $\hat{\mu}_i(u)$ to the observed responses $\bar{R}_i(u)$. We then assume the background to be distributed as

$$R_i^{\text{bg}}(u) \sim \mathcal{N}(\hat{\mu}_i(u), \sigma_{\text{bg}}^2), \tag{6}$$

where σ_{bg} is a global parameter in raw ToF units, typically in the range of a few hundred units.

The full model $P(Y_t|X_t)$ is now the composition between the background responses R_i^{bg} and the object (foreground) responses R_i^{obj}. For a given object hypothesis X_t the renderer can perform this composition easily as it computes a mask of object locations during rendering. Let use denote the mask by $M(u) \in \{0, 1\}$ where $M(u) = 1$ denotes a location where the object hypothesis causes the location u to be occupied. We obtain the full model as

$$R_i(u) \sim \begin{cases} \mathcal{N}(R_i^{\text{obj}}(u), \sigma_{\text{obj}}^2), & \text{if } M(u) = 1, \\ \mathcal{N}(\hat{\mu}_i(u), \sigma_{\text{bg}}^2), & \text{otherwise.} \end{cases} \tag{7}$$

Here the additional parameter σ_{obj} is a constant specifying the assumed noise in the object responses. From (7) and assuming independent pixels we see that the full raw ToF frame is modeled by a product of Gaussian distributions, hence itself is a multivariate Gaussian. Therefore we compute the log-likelihood function $\log P(Y_t|X_t)$ as

$$\begin{aligned} \log P(Y_t|X_t) = &- \sum_{u:M(u)=1} \left[\frac{(\bar{R}_i(u) - R_{\text{obj}}(u))^2}{2\sigma_{\text{obj}}^2} + \log \sigma_{\text{obj}} \right] \\ &- \sum_{u:M(u)=0} \left[\frac{(\bar{R}_i(u) - \hat{\mu}_i(u))^2}{2\sigma_{\text{bg}}^2} + \log \sigma_{\text{bg}} \right] + C, \end{aligned} \tag{8}$$

where $C = -\frac{n}{2}\log(2\pi)$ is a constant independent of the observation ($n = 512 \cdot 424$ is the sensel count in a frame), and can be omitted.

4. Implementation and Validation

Implementation details We implement the above modifications to the Kinect sensor [2] by using a custom firmware and modified driver software. However, in principle, a similar mode of operation could be supported by other phase-based ToF cameras available commercially, *e.g.* the ones available from PMD, Intel, and Mesa Imaging.

The tracker is implemented in C++ on a CPU and the rendering and likelihood computations are performed entirely on the GPU through custom shaders. A tiled layout enables to evaluate over 8000 particles in parallel and when using 4096 particles allows real time tracking at 300 Hz.

Experimental setup A. For our experiments we use two different setups as shown in Fig. 4 and Fig. 5. In the first setup, we use a static camera mounted on a tripod and a table tennis ball as object model. The ball is released from a fixed position with no inertia and falls downwards driven purely by acceleration due to gravity. Quickly the ball reaches a velocity that prevents a reliable depth reconstruction using phase unwrapping because of insufficient overlap in the individual raw ToF frames (see §5.1). Therefore there

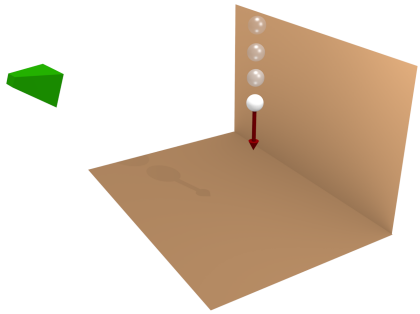

Figure 4. **Experimental setup A.** A table tennis ball is released from a stationary position and accelerates towards the ground. The camera observes this fall at a slightly downwards angle.

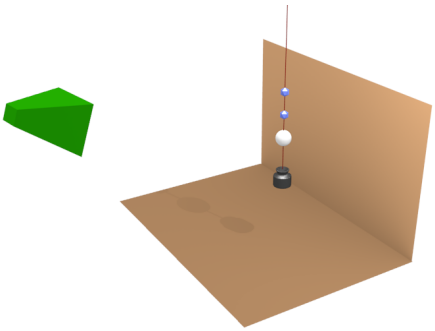

Figure 5. **Experimental setup B.** A table tennis ball attached to a rope swings as a pendulum. Attached to the rope are two reective markers used for motion capture. The rope is kept straight using an attached weight. The scene is captured by both the Kinect camera and a commercial motion capture system consisting of 11 cameras (not shown).

is no ground truth depth available. To nevertheless assess tracking performance quantitatively we use the following procedure. Because the ball starts from rest, the trajectory lies on a line in 3D space. Our tracker predicts object coordinates at each observation as the average of the weighted particle positions. For each sequence we use the predicted coordinates and fit a line using least squares. Each predicted coordinate deviates from the line and the magnitude of this deviation is a reasonable proxy of the quality of the tracking result.

Specifically, because motion is present only in the y/z plane, we fit a least squares regressor $z_t \approx ay_t + b$ from the the object position at step t as predicted by the tracker. The error metric is then the root mean squared error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1:T}(z_t - (ay_t + b))^2}. \qquad (9)$$

To avoid potential biases due to initialization effects we perform the above for only the second half of each sequence. All our methods use the same transition model and

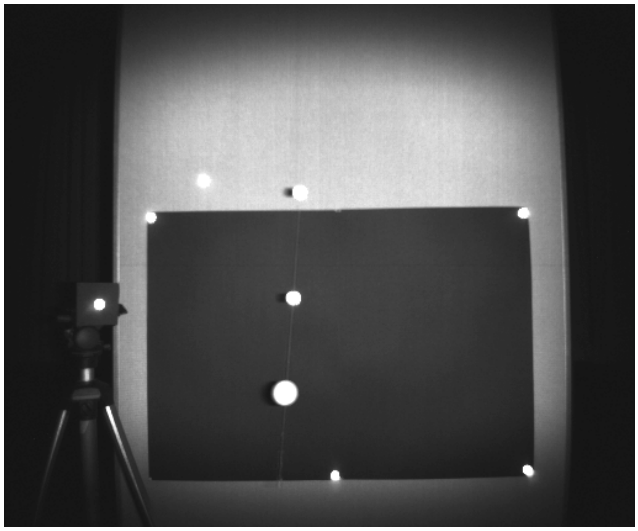

Figure 6. **Experimental setup B.** A table tennis ball is attached to a rope with two reflective markers. Additional markers in the scene allows us to register the coordinate frames of the camera and of the motion capture system. Shown in an averaged infrared image of the ToF camera.

the same parameters in (5), so no systematic bias due to assuming a strict linear motion benefits one model over the other.

Experimental setup B. In our second setup, the table tennis ball is attached to a rope together with two reflective markers. The markers are tracked in 3D at 150 Hz using an eleven-camera motion capture system (Qualisys QTM, Qualisys Inc., Gothenburg, Sweden). Attached to the end of the rope is also a weight, which straightens the rope. For quantitative comparison we transform the 3D trajectory of the motion capture system to the Kinect camera coordinate frame and compute the root mean squared error (RMSE) between the three dimensional coordinates of the raw ToF tracker result and the motion capture system output. To achieve valid ground truth for the different exposure timings of the Kinect V2 camera, we linearly interpolate positions on the motion capture trajectory based on the timestamps of the Kinect and use these interpolated positions as ground truth for the evaluation.

Notes on the accuracy: The motion capture system is calibrated with a residual of less than 2 mm. The coordinate frames of the Kinect camera and the motion capture system are registered by using six reflective markers which are visible in the Kinect camera frame. In the Kinect camera frame, depth values are assigned to the markers using the standard depth reconstruction of the Kinect. Registration of both coordinate frames is achieved by using Kabsch's algorithm [14] with a residual of $8-9$ mm. Due to the remaining minor systematic deviations between the commercial mo-

		Camera frame rate	
		Low fps	High fps
Mode	unmodulated	video camera	highspeed camera
	modulated	Kinect V2 depth	**our Kinect V2**

Table 1. Relevant dimensions of camera operation and frame rate for object tracking. Within the four quadrants going to the right or going downwards potentially improves tracking performance. Our approach combines the benefits of a high frame rate with phase modulation to provide superior tracking performance.

tion capture system and our tracking system, we apply a scaling and translation transformation to register the trajectories. The transformation requires estimating six parameters and the typical trajectory length is 1350 measurements.

5. Experiments

Our approach combines the use of raw ToF observations with the use of a high frame rate. These benefits are complementary, and are best understood as part of a landscape of possible camera modes, as shown in Table 1. As a consequence we design the experiments to verify that both the ToF modulation and the equispaced frames are beneficial for tracking and that these benefits are complementary.

We first demonstrate that tracking based on the Kinect V2 depth reconstruction fails for fast moving objects because of motion artifacts.

5.1. Failure of Standard Kinect Depth Tracking

The underlying assumption of the Kinect V2 depth reconstruction algorithm is a static scene. If an objects moves between two raw frames that is then used for reconstructing the depth frame, artifacts become visible in the depth reconstruction. Fig. 7 shows an overlay of the raw frames of the falling table tennis ball together with the depth reconstructions obtained from these frames. A depth value can only be reconstructed in those regions where the moving object overlaps in all the raw frames used for the depth reconstruction. Therefore it is obvious that the strategy of first reconstructing a depth image and then tracking fast moving objects has to fail. We therefore propose to track the object in three dimensions by directly using the raw data of the sensor.

5.2. Tracking with Raw ToF Observations

We now show that the unknown depth of an object can be obtained by our model-based tracking method. Fig. 8 depicts the estimated depth with respect to the vertical y coordinate of the table tennis ball. To validate our method, we show for comparison the reconstructed depth values of the table tennis ball, measured in the depth frame in the overlapping region (compare to Fig. 7).

The exposures of the camera are all at the beginning of each 30Hz depth frame capture in order to minimize motion artifacts. For tracking purposes these unevenly spaced exposures are suboptimal: the larger time-gap between the captures leads to low quality of the depth estimate for the first frame of each capture, as is visible by the sharp drop of estimated depth every 9th frame in Fig. 8.

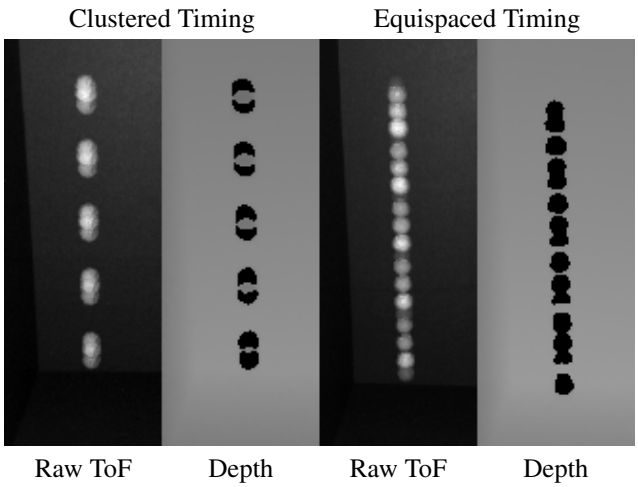

Figure 7. **Depth reconstruction failure (Experiment A).** The ball quickly reaches a velocity that prevents a successful depth reconstruction. This is due to insufficient overlap of the object in the nine frames used for reconstruction. Left two images: an overlay of the raw captures for five frames and the corresponding depth reconstructions, using the standard 'clustered' exposure timing of the Kinect. Note the depth reconstruction gets worse with increasing velocity (the black 'holes'). Right two images: Equidistant exposure timing. Depth reconstruction now completely fails. However, we show that this timing is beneficial for our proposed tracking method because we can directly leverage the high frame-rate raw ToF information.

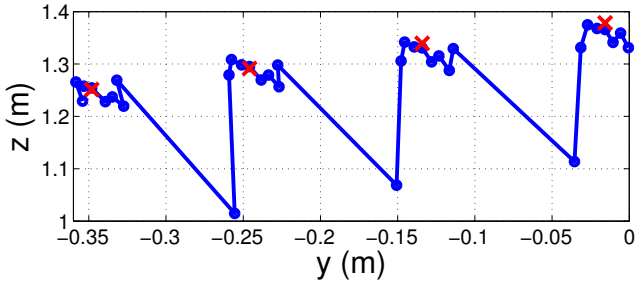

Figure 8. Validation of the estimated depth values when tracking off raw ToF images captured using the standard 'clustered' temporal spacing (blue), in comparison to the depth values of the standard time of flight reconstruction method (red). (Experiment A)

5.3. Benefit of Equispacing

To overcome the large time gap between the 30Hz captures we propose to use an equidistant timing of the exposures. Fig. 9 clearly demonstrates that this increases the stability of the trajectory estimates. Note that Fig. 9 tracks a

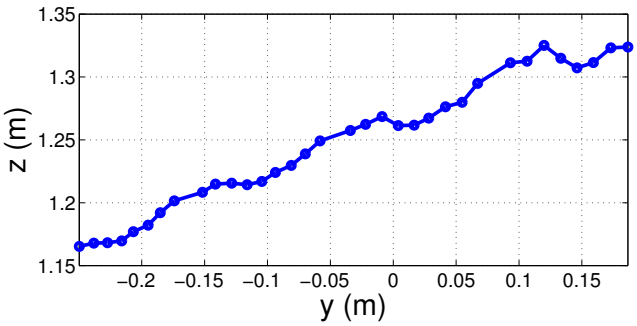

Figure 9. Equispaced exposure timing leads to a more stable depth estimate when tracking from raw ToF captures. (Experiment A)

different sequence to Fig. 8, since due to interference we are unable to capture simultaneously with both clustered and equidistant.

We also quantitatively compared the effect of the equidistant timings in comparison with the standard clustered exposure timings by computing the root mean squared error of the residual towards a linear regressor as explained in the beginning of this section. We compared both exposure timings by individually tracking 10 sequences with 60 frames each of a table tennis ball falling straight to the ground. The camera is slightly tilted downwards which leads to a linear relation between the y and the z coordinate. Our results in Table 2 and Table 3 show first that our model-based tracking method is highly accurate and second that the equidistant based shutter profile further improves this accuracy for motion in the x- and y-coordinate. The overall accuracy (RMSE) mainly depends on the object's speed whereas the root mean squared errors for the individual coordinates show that the equidistant exposure timing improves the accuracy for tracking the objects motion in the x- and y- coordinates. The clustered exposure timing minimizes the distance between the different captures and allows for a better depth reconstruction of a fast moving object. For a slow moving object however, the equidistant exposure timing allows to reconstruct the depth with an even higher accuracy.

		RMSE
Exposure Timing	clustered	16.2 mm
	equidistant	15.9 mm

Table 2. **Experiment A.** Quantitative comparison of the standard clustered exposure timing and our proposed equidistant timing for a table tennis ball accelerated by gravity.

Exposure Timing	Object Speed	RMSE	RMSE [mm]		
			x	y	z
clustered	1.33 km/h	16.3 mm	4.3	2.3	15.5
	2.12 km/h	19.3 mm	8.4	9.1	14.7
equidistant	1.01 km/h	15.5 mm	4.5	3.3	14.4
	2.32 km/h	23.7 mm	6.4	6.9	21.8

Table 3. **Experiment B.** Quantitative comparison of the different exposure timings for a slow and fast moving table tennis ball. Shown is the root mean squared error between the raw ToF tracker and a commercial motion capture system, for all three coordinates and separately for the x-, y- and z-coordinate in the camera coordinate frame.

6. Discussion

6.1. Tracking from Raw ToF

A potential benefit of using (8) to directly fit to the raw ToF observations (as opposed to fitting to reconstructed depth images) is a reduced computational cost: whereas a depth-based tracker first has to reconstruct depth and compute a likelihood function based on this estimated depth, we can skip the depth reconstruction altogether. While the depth reconstruction in the Kinect device and drivers is optimized and can be highly parallelized, this computational saving could be significant for mobile, power-limited devices.

But an even greater benefit of fitting to raw ToF observations is that we avoid the artifacts that would result from attempting to reconstruct depth from observations of fast motion (examples of motion that would cause problems are shown in Fig. 2). Each raw ToF frame has an order of magnitude shorter exposure time compared to the full sequence of nine frames required for a depth reconstruction, even when clustered as in the standard Kinect.

6.2. Equispaced ToF Captures

The Kinect V2 camera supports a flexible scheduling of frame capture times and we can space the raw captures uniformly over time (see Fig. 1). When the goal was depth reconstruction, it made sense to cluster the frames in time to minimize motion artifacts. But in our approach, we do not need to reconstruct depth, and so are free to space the captures uniformly. Capturing frames more uniformly is especially useful if the object's movement in the x- and y- coordinates, the space of the image plane, is of higher importance. The tracking accuracy in these coordinates is directly improved by the more uniformly spaced captures. However, because the object's movement between the individual captures is higher in comparison to the standard timing, reconstructing the depth of the object becomes more difficult which results in a slight decrease of accuracy in the estimated depth.

We are still exposed to intra-frame motion artifacts, but these are negligible compared to the motion artifacts present in the original stack of frames because the exposure time of a single frame is an order of magnitude smaller than the exposure time across all ToF frames required for depth reconstruction.

6.3. Limitations and Future Work

As we have shown, the ToF tracking framework obtains several benefits, but there are also some limitations in practice. Tracking in ToF captures of a cluttered scene is much harder than tracking in the depth frame. As future work, we hope to explore other model-based tracking algorithms which allow for tracking more general objects including articulations, as well as arbitrary backgrounds. Also, especially for articulated objects, we believe that the benefits of both methods can be combined by using the more robust depth based tracking for slow moving object parts and utilizing the higher time resolution of the ToF tracking framework for the faster moving parts.

7. Conclusion

We proposed a novel mode of operation of a commonly available ToF sensor for the purpose of high-speed object tracking. Through experiments we demonstrated improved tracking accuracy due to two distinct contributions: modeling raw ToF frame observations, and spacing the capture uniformly over time.

Our approach is potentially useful for other fast moving objects, for example the human hand. The approach will likely struggle when tracking objects at large distances (> 10m) because the active illumination does not extend to this range; however, one could envision an extended system which falls back to ambient light for large distances, only leveraging the ToF illumination information for accurate tracking at shorter distances.

We believe our work is just the first step in adapting ToF sensor operation to better fit computer vision tasks; we considered tracking, but other vision applications such as surface reconstruction and camera localization may benefit similarly. In addition, whereas we still operate the camera in an manner that is fixed apriori, it is conceivable to further adapt the camera operation online given observed data.

Acknowledgements The authors would like to thank the chair of Information-oriented Control (TUM) for their support with the motion capture system. This research was supported by the ERC Starting Grant "ConvexVision" and the Microsoft Research internship program.

References

[1] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011. 1

[2] C. S. Bamji, P. O'Connor, T. A. Elkhatib, S. Mehta, B. Thompson, L. A. Prather, D. Snow, O. C. Akkaya, A. Daniel, A. D. Payne, T. Perry, M. Fenton, and V.-H. Chan. A 0.13 μm CMOS system-on-chip for a 512 $\times$ 424 time-of-flight image sensor with multi-frequency photo-demodulation up to 130 MHz and 2 GS/s ADC. *J. Solid-State Circuits*, 50(1):303–319, 2015. 2, 4

[3] J. F. Blinn. Models of light reflection for computer synthesized pictures. *ACM SIGGRAPH Computer Graphics*, 11(2):192–198, 1977. 4

[4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell*, 25(5):564–575, 2003. 1

[5] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009. 3

[6] J. Durbin and S. Koopman. *Time Series Analysis by State Space Methods: Second Edition*. Oxford Statistical Science Series. OUP Oxford, 2012. 3

[7] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, UAI'97, 1997. 4

[8] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993. 3

[9] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin. Phasor imaging: A generalization of correlation-based time-of-flight imaging. Technical report, Department of Computer Science, Columbia University, 2014. 2

[10] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Real-time camera tracking: When is high frame-rate best? In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII*. Springer, 2012. 1

[11] M. E. Hansard, S. Lee, O. Choi, and R. Horaud. *Time-of-Flight Cameras - Principles, Methods and Applications*. Springer Briefs in Computer Science. Springer, 2013. 2

[12] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 3

[13] M. Isard and A. Blake. A smoothing filter for CONDENSATION. In H. Burkhardt and B. Neumann, editors, *Computer Vision - ECCV'98, 5th European Conference on Computer Vision, Freiburg, Germany, June 2-6, 1998, Proceedings, Volume I*, volume 1406 of *Lecture Notes in Computer Science*, pages 767–781. Springer, 1998. 3

[14] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976. 5

[15] D. Koller, K. Daniilidis, and H. H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, June 1993. 3

[16] M. Kristan, R. P. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojír, G. Fernández, A. Lukezic, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangersic, G. Häger, F. S. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Y. Choi, J.-W. Choi, J. F. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Öfjäll, K. M. Yi, L. Qin, L. Wen, M. E. Maresca, M. Danelljan, M. Felsberg, M.-M. Cheng, P. H. S. Torr, Q. Huang, R. Bowden, S. Hare, S. Y. Lim, S. Hong, S. Liao, S. Hadfield, S. Z. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, and Z. H. Niu. The visual object tracking VOT2014 challenge results. In *ECCV Workshops*. Springer, 2014. 1

[17] C. H. Lampert and J. Peters. Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components. *Journal of Real-Time Image Processing*, 7(1):31–41, 2012. 1

[18] R. Lange and P. Seitz. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics*, 37(3):390–397, 2001. 2

[19] D. Lefloch, R. Nair, F. Lenzen, H. Schäfer, L. Streeter, M. J. Cree, R. Koch, and A. Kolb. Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging: Sensors, Algorithms, and Applications*, pages 3–24. Springer, 2013. 2

[20] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno. 1ms column parallel vision system and it's application of high speed target tracking. In *ICRA*. IEEE, 2000. 1

[21] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*. Springer, 2004. 1

[22] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 1

[23] R. Schwarte, Z. Xu, H.-G. Heinol, J. Olk, R. Klein, B. Buxbaum, H. Fischer, and J. Schulte. New electro-optical mixing and correlating sensor: facilities and applications of the photonic mixer device (PMD). In *Proc. SPIE*, volume 3100, pages 245–253, 1997. 2

[24] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999. 4

[25] A. D. Worrall, R. F. Marslin, G. D. Sullivan, and K. D. Baker. Model-based tracking. In P. Mowforth, editor, *BMVC*, pages 1–9. BMVA Press, 1991. 3

[26] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011. 1

[27] A. Yilmaz, O. Javed, and M. Shah. Object tracking: a survey. *ACM Computing Surveys*, 38(4):13:1–13:45, Dec. 2006. 1

[28] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *ICPR*, 2004. 4